ЛЕКЦІЯ 1

Retrieval-Augmented Generation

Проблема генеративних моделей та ідея RAG

LLM • Обмеження • Векторний пошук • Архітектура RAG

Обмеження великих мовних моделей

Чому LLM потребують зовнішньої пам'яті

Hallucinations

Модель «вигадує» факти з впевненим тоном. Статистичний прогнозатор, а не база знань.

Context Window

Неможливо «вгодувати» цілий датасет за один запит — є ліміт токенів.

Knowledge Cutoff

Знання моделі обмежені датою навчання. Нові дані недоступні без RAG.

Lost in the Middle

Важлива інформація в середині великого контексту часто ігнорується.

Немає логіки

Погано справляється з математикою та приватними даними без інструментів.

Висока вартість

Запуск топових API для мільйонів запитів — дорогий ресурс у 2026 році.

Ідея Retrieval-Augmented Generation

Класична LLM

$$P(y \mid x)$$

Відповідь залежить лише від запиту.
Немає зовнішніх знань.

RAG-модель

$$P(y \mid x, D)$$

D = {d1..dk} - k релевантних документів із зовнішньої бази знань.

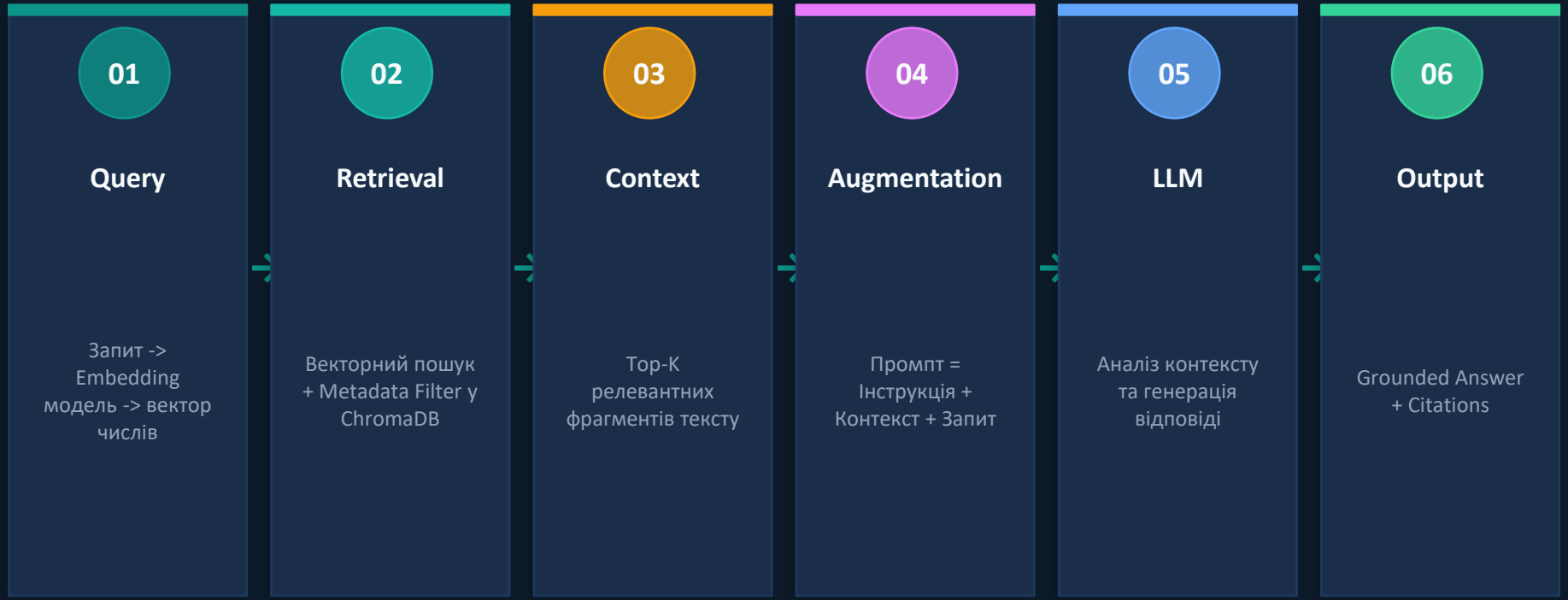
1. Retrieval — пошук контексту

D = Retrieve(x)
Знаходить k релевантних документів з бази за допомогою векторного пошуку.

2. Generation — генерація відповіді

y = Generate(x, D)
LLM отримує запит + документи та формулює точну відповідь.

Архітектурний конвеєр RAG



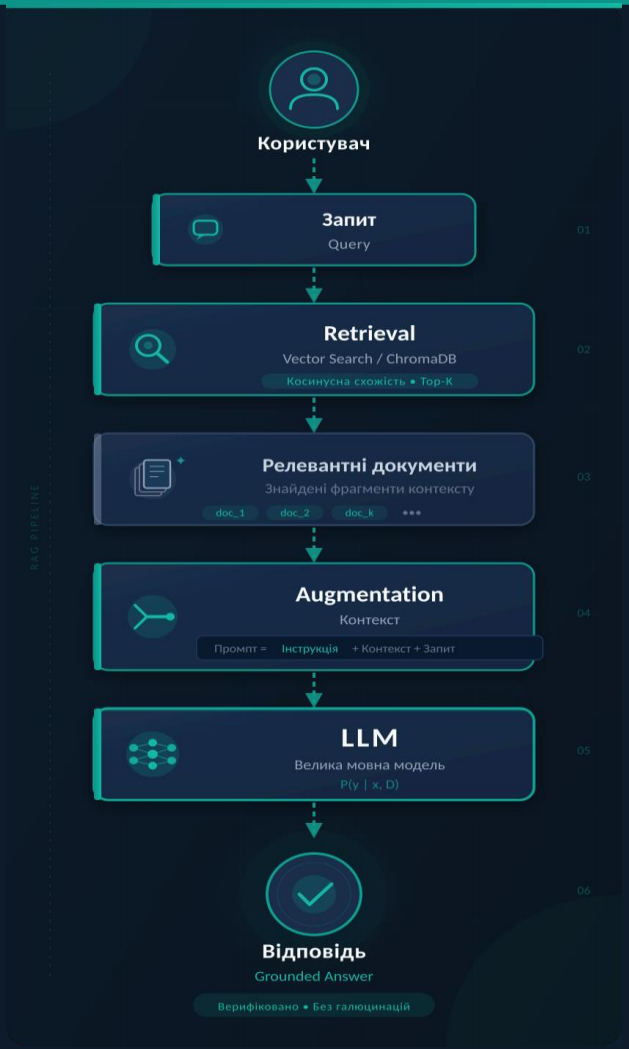
Retrieval — точність | Augmentation — контекст | LLM — мова відповіді

Загальна схема RAG

Retrieval
Відповідає за точність та актуальність даних.

Augmentation
Відповідає за контекстуалізацію (підготовку інструкцій).

LLM
Відповідає за зрозумілість та мову відповіді.



Види RAG

Naive RAG

- Текст -> Чанки -> Ембединги
- Пошук Top-K -> Генерація
- Лінійна схема без зворотного зв'язку
- Обмеження: нерелевантний пошук -> помилка

Advanced RAG

- Pre-retrieval: Query Expansion, Sub-queries
- Post-retrieval: Reranking (Cross-Encoder)
- Prompt Compression — видалення зайвого
- Підвищена точність пошуку

Modular RAG

- Hybrid Search: вектори + BM25 (ключові слова)
- Підключення зовнішніх джерел (Google/Bing)
- Вільна комбінація модулів
- Найвища адаптивність

Тренди: GraphRAG | Agentic RAG | Long-Context RAG

Grounded Answers

Відповідь суворо базується на зовнішніх даних — антипод галюцинацій

Зв'язок із джерелом

Відповідь містить посилання на конкретні документи з ChromaDB.

Без галюцинацій

Якщо відповіді немає — модель чесно каже «Я не знаю».

Верифікованість

Користувач може перевірити кожен фрагмент-джерело відповіді.

Grounded vs Hallucination

Запит: «Коли вийшов Інтерстеллар?»

Grounded Answer

«Згідно з датасетом Netflix, фільм вийшов у 2014 році [Джерело: рядок 452].»

Hallucination

«Фільм вийшов у 2015 році.» — модель використала старі ваги замість ваших даних.

Метрика: Faithfulness (RAGAS) — перевірка кожної тези на відповідність контексту

Ключові висновки

- LLM — статистичний прогнозатор токенів, а не логічна база знань. Це першопричина галюцинацій і knowledge cutoff.
- RAG розширює LLM зовнішньою пам'яттю: P(y | x, D). Відповідь умовлюється знайденими документами.
- Конвеєр RAG: Query -> Embedding -> Vector Search -> Augmentation -> LLM -> Grounded Answer.
- Naive RAG — простий і лінійний. Advanced RAG — точніший (Reranking). Modular RAG — найгнучкіший.
- Grounded Answers верифіковані та прив'язані до джерел. Метрика Faithfulness (RAGAS) оцінює якість.